

Anti-aliasing

Modern computer games are beautifully rendered. Tom Royal looks at the technique that provides smooth onscreen lines

The first 3D game for a home computer was called 3D Monster Maze. Released in 1981, it involved running around a 3D maze attempting to find the exit while avoiding a 2D Tyrannosaurus Rex that wanted to eat you. Running on a Sinclair ZX81 home computer, 3D Monster Maze ran at around six frames per second (fps). Today's 3D games are obviously more advanced, but they still suffer from one of the fundamental problems evident in 3D Monster Maze.

A computer display consists of a large grid of coloured pixels. This grid pattern, with pixels placed in horizontal and vertical rows, allows a display to show some objects, such as a coloured box with perfectly vertical and horizontal lines, very sharply. Each pixel is either coloured, if it is part of the box, or uncoloured if it is not. However, displaying lines that aren't aligned with the rows of pixels in the screen is more difficult.

In early games, the blocks of black and white used to make up each picture turned diagonal lines into a shape akin to a staircase. Today's games use higher resolutions so the effect is less pronounced, but you can still notice jaggedness – or aliasing – around the edges of objects.

SUPER-SAMPLING

With a black and white display like the ZX81's there was no way to reduce aliasing. When using a greyscale or colour display, though, it can be drastically reduced. If a black diagonal line is drawn on a white background, for instance, the edges can use shades of grey.

The simplest method for performing this kind of anti-aliasing is called super-sampling. This works by making several different colour calculations for every pixel to be produced, and averaging them. Two methods are common: ordered and rotated grid.

In a non-anti-aliased picture, the colour of each pixel is determined by whether or not it is half-filled by the foreground object. If it is, it's given the colour of the object. If it isn't, it's given the background colour. To determine whether it is half-full or not, a sample is taken at the centre point of the pixel. If this pixel is found to be obscured by the foreground object, say a green leaf, the whole pixel will be painted green. If it isn't, the pixel will be given the background colour.

If using ordered grid super-sampling (OGSS), this single sample is replaced by several that are placed around the pixel in a manner similar to the arrangement of dots on the side of a die. If using 2x OGSS, one sample is taken from the bottom left of the pixel and one from the top right. If using 4x OGSS, one sample is taken from each quadrant of the pixel. Each sample is tested to see what object is in view at that point, and each sample is converted into a colour by applying the texture of the object visible at that point. Any pixel shaders used to produce effects such as lighting are also run for each sample. Once all four samples have been converted into a colour, the colours are averaged. This final colour is used for the final pixel displayed onscreen. If drawing the edge of a black line on a white background, this process produces shades of grey that give the edge of the line a smoother appearance.

Rotated grid super-sampling (RGSS) works similarly. The difference lies in the position of the samples used when calculating each pixel. With RGSS, the patterns of two, four or more samples are rotated within the pixel by 25° to 30° so that no two samples are on the same horizontal or vertical line. This slight change in sample position produces better results when displaying edges that are nearly horizontal or nearly vertical.

OGSS and RGSS smooth the appearance of the entire 3D image effectively. The problem with both methods is performance. Using either kind of 4x super-sampling involves four times as many calculations as using no anti-aliasing, as well as more buffer memory. If a game is running at a resolution of 1,024x768 pixels and at 60fps, 4x super-sampling requires 188,743,680 sets of calculations per second. For this reason, super-sampling has in effect been abandoned in favour of less-intensive methods that produce similar results.

MULTI-SAMPLING

All modern 3D cards use a technique called multi-sample anti-aliasing for most of their anti-aliasing. This method is similar to super-sampling, but requires far fewer calculations per pixel. As with super-sampling, a number of points within the pixel are sampled. These can be in an ordered grid pattern or a rotated grid pattern, as used in modern nVidia

FURTHER READING

Explanation of multi-sampling
www.ati.com/products/pdf/smoothvision.pdf

Transparency AA explained
http://developer.nvidia.com/object/transparency_aa.html

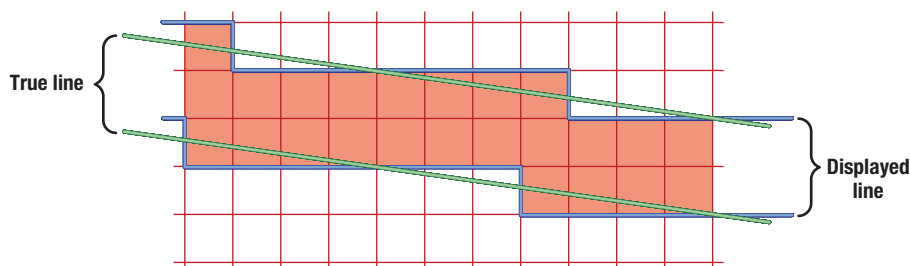
INTRODUCTION

Consider a coloured diagonal line passing through a grid of pixels. Some pixels will be completely covered by the line, some are mostly covered and others are only slightly touched at their edges. A computer cannot partially fill a pixel; each pixel on the display is either coloured or not. Consequently, the computer bodes it. Those pixels that are more than half-filled by the line are displayed as full, and those less than half-filled are displayed as empty.

The result is a line that looks similar to the original, but its edges are not smooth. Instead, they are covered in a repeating pattern of steps. This stepping effect is known as aliasing.

By using a darker grey for pixels that should be mostly filled by the black line and a light grey for those that should contain a small amount of black line and a large amount of white background, the line will look far smoother onscreen. This kind of technique is known as anti-aliasing. Most modern 3D graphics cards support several types of anti-aliasing.

The aliasing problem



▲ With no anti-aliasing, a diagonal line passing through a grid pixels will have jagged edges, as each pixel is either coloured or not coloured

Tom Royal
Labs Editor



tom@computershopper.co.uk

graphics cards. Modern ATI graphics cards use a programmable sample pattern.

Each one of the sample points is tested to see which onscreen object is closest according to its Z-buffer value, which measures depth. If a nearby tree overlaps the pixel, the value will be close to 0, while if a distant mountain fills the pixel the value will be nearer to 1. The computer works out which object, and which texture, is seen at each point. Unlike in a super-sampling system, however, individual texture and pixel shader operations are not performed on these samples. Instead, one single set of these operations is executed based on a separate sample point. This point can be the centre of the pixel, or in a more advanced system the 'centroid', the centre of the foreground shaded area. The samples that were found to contain the foreground object are given this colour, and the others are given the background colour. The multiple sample values are then averaged to create the final pixel colour.

Multi-sampling is effective at smoothing jaggedness. It requires a larger amount of Z-buffer memory to store the increased amount of depth information it requires, but it is significantly less demanding than super-sampling as the number of texture and shader operations required is far smaller. Its weakness lies in the use of only one set of pixel shader operations per final pixel. As these pixel shaders include the alpha test, which checks for transparencies, multi-sampling performs poorly on textures that combine transparent and visible areas.

STATE OF THE ART

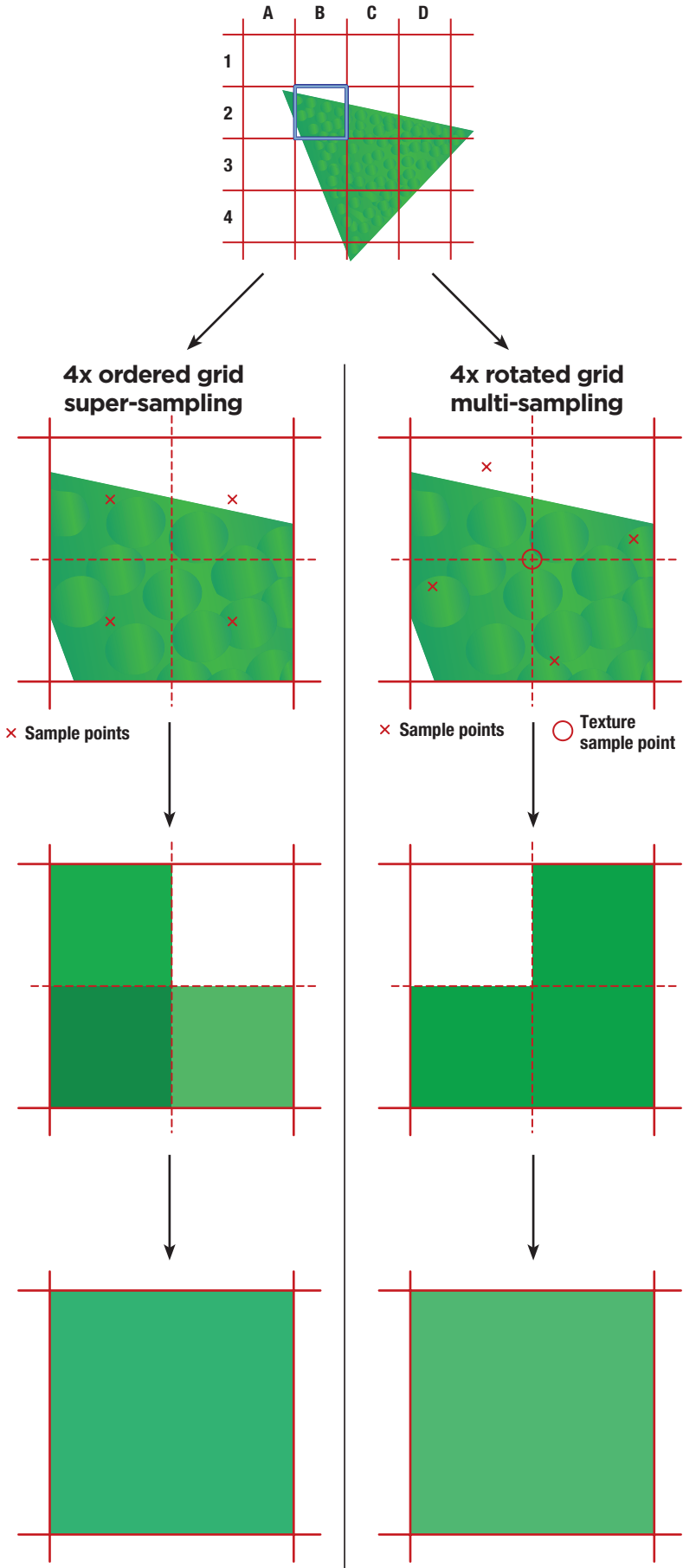
This problem with multi-sample anti-aliasing on a partially transparent texture is most obvious in games that use objects such as shrubbery and wire fences. These combine areas of transparency, such as the gaps in the fence, with visible areas, such as the wires. As multi-sample anti-aliasing does not take account of the detail inside textures, the areas where these transparent and opaque parts of the texture meet suffer badly from aliasing. Only relatively recent graphics cards have the ability to combat this problem.

ATI and nVidia have developed anti-aliasing techniques that combine multi-sample anti-aliasing for the entire image with further anti-aliasing applied to textures that include transparent areas. ATI calls its technique Adaptive Anti-Aliasing, while nVidia refers to its as Transparency Anti-Aliasing. Adaptive Anti-Aliasing was introduced with ATI's X1000 series of graphics processors, and Transparency AA is available on nVidia's GeForce 7 series processors.

The principle of both systems is the same. During processing, the anti-aliasing software attempts to detect textures that have a mix of transparent and visible areas. The entire screen is anti-aliased using the usual high-performance multi-sampling method, but the mixed areas are subjected to a second anti-aliasing using a different method. ATI's Adaptive Anti-Aliasing uses a form of super-sampling where the area is rendered several times. Each pass is slightly offset by less than a pixel, and the multiple offset pixels are averaged to produce the final pixel shade.

NVidia's driver offers a choice of two types of anti-aliasing for semi-transparent textures. It can use super-sampling, producing very good image quality, or a mode that nVidia calls Transparency Adaptive Multi-Sampling. This second mode works similarly, but using only one texture sample per pixel. It's faster, but less effective. 

Super-sampling v multi-sampling



▲ 4x OGSS (left) uses four sample points taken from each quadrant of a pixel. Each sample is then converted into a colour, and the four colours are averaged to produce the final colour. The more efficient 4x RGMS rotates the sample points and saves processing by using a single texture sample